Universidad del valle de Guatemala

Algoritmos y programación básica

Sección: 90

**Investigación corta**

Alumnos:

Diego Renato Pérez Bercian carne 12142

Iván Avidal Concua Lara carne

Pablo José Chavac Figueroa carne 12331

Fecha de entrega: 10/04/2012

**ANALISIS DE REQUERIMIENTOS**

1. **PREGUNTAS INSUSTITUIBLES**

* **¿Cuál es la meta?**

R. La meta es poder simular un juego de elevadores el cual, muestre en pantalla la posición de 5 elevadores, lo cuales deben moverse hacia arriba y hacia abajo, hasta que todos se encuentren en el intervalo deseado.

* **¿Cuáles son las salidas?**

R. Dependiendo de las entradas, las salidas pueden ser: mensaje de error a la hora de seleccionar un numero o identificación de elevador que no es permitido; un mensaje de error cuando los elevadores llegan a su máximo o mínimo permitido; también cada turno enseña el número de movimientos que ha hecho el jugador; al cumplir con todas las condiciones el juego se finaliza y muestra en pantalla el número de movimientos hechos y pregunta si desea volver a jugar.

* **¿Cuáles son las entradas?**

R. La selección de elevadores a mover. La opción de subir o bajar dos elevadores distintos. La opción de iniciar el programa, así como reiniciarlo.

* **¿Cuáles son las condiciones/restricciones?**

R. El usuario no debe ingresar opciones no validas como texto en vez de números o opciones invalidas, que puedan generar un error. Los elevadores no deben poder superar el nivel 49 y no deben bajar más del nivel 0.

1. **ALGORITMO NARRATIVO**

INICIO

Impresión en pantalla de los niveles de los 5 elevadores

Mientras los 5 elevadores no se encuentren en el intervalo de [21,25]

Elección de dos niveles

Elegir subir 8 niveles o bajar 13 niveles

Si elige subir

Si los elevadores son válidos y no superan los 49 niveles del edificio

Sumar 8 niveles a los elevadores ingresados

Sumar al contador 1 unidad

De lo contrario

Mostrar en pantalla mensaje de error indicando el porqué de este.

Se regresa a la elección de niveles

Si elige bajar

Si los elevadores son válidos y la resta de niveles no es menor que 0

Restar 8 niveles a los elevadores ingresados

Sumar al contador 1 unidad

De lo contrario

Mostrar en pantalla mensaje de error indicando el porqué de este.

Se regresa a la elección de niveles

Mostrar un mensaje, “Ganaste”, que muestre el número de intentos

Preguntar si desea volver a jugar

Si desea volver a jugar

Regresar al inicio

Si no desea volver a jugar

FIN

**DISEÑO**

1. **¿Cómo llevar a cabo la solución? (variables y objeto a utilizar)**

Para poder simular el juego se deberán utilizar varia variables que representan a cada elevador, un contador para mostrar el número de intentos y una variable que dejara que el ciclo continúe o se interrumpa. Una variable temporal que capture los niveles a modificar, y otra la opción a realizar

* Variables:
* el1- Esta variable representa al elevador 1 y se encuentra en el nivel 17
* el2- Esta variable representa al elevador 1 y se encuentra en el nivel 26
* el3- Esta variable representa al elevador 1 y se encuentra en el nivel 20
* el4- Esta variable representa al elevador 1 y se encuentra en el nivel 19
* el5- Esta variable representa al elevador 1 y se encuentra en el nivel 31
* e1-Almacena el número del primer elevador elegido
* e2-Almacena el número del segundo elevador elegido
* e3-Almacena la opción elegida de subir o bajar
* y-almacena la opción de iniciar el ciclo
* x-Es el contador de intentos del juego
* Objetos:
* Uno de los objetos eran los elevadores, que se encontraban en sus respectivos niveles, estos debían moverse de arriba hacia abajo.
* Otro objeto importante a tomar en cuenta era el edificio de 49 pisos por los cuales se debían desplazar los elevadores.

1. **FUNCIONES DESARROLLADAS POR EL PROGRAMADOR**

* **Subir (num1,num2):** En esta función se verifica si los dos elevadores al subir no sobrepasaran los 49 niveles y luego se pasa a sumar 8 unidades al nivel que está siendo ingresado. Esta función recibe dos valores, pero solamente regresa uno, para ser usada correctamente debe ser utilizada dos veces dentro de una condición.
* **Bajar (num1,num2):** En esta función se verifica si los dos elevadores al bajar 13 niveles no resultara en una cantidad menor que cero, luego resta 13 unidades al elevador ingresado de primero. Esta función al igual que la primera recibe dos valores de entrada pero solo regresa uno de salida.
* **Salir():** Esta función es utilizada únicamente para salir del programa utilizando la función exit().

Inicio

Los elevadores con los niveles en los que están.

Mientras los 5 elevadores no se encuentren en el intervalo de [21,25]

F

V

Elija los dos elevadores:

Elegir subir 8 niveles o bajar 13 niveles

6

3

2

6

3

Si elije subir

Error

Error

Restar 8 niveles a los elevadores ingresados

Sumar al contador 1 unidad

Si los elevadores son válidos y la resta de niveles no es menor que 0

Si elije bajar

Sumar 8 niveles a los elevadores ingresados

Sumar al contador 1 unidad

los elevadores son válidos y no superan los 49 niveles del edificio

F

V

F

V

5

4

4

Ganaste

H=¿Desea volver a jugar?

Si H es no

Si H es si

fin

5

**CONCLUSIONES**

* Es importante tener organización en el código del programa para a la hora de hacer un cambio que sea necesario, no sea confuso, ni dañe el resto del programa.
* Se vio que el uso de funciones, ayuda a economizar espacio en el programa original, además de que reducía el margen de error, porque no era necesario reescribir los procesos similares.
* Los repetidores son la base del programa ya que son la estructura que permite ejecutar el programa tantas veces como sea necesario para poder cumplir con los requerimientos del juego.
* En el desarrollo del programa se pudo observar que es necesario tener en cuenta todos los posibles errores y/o acciones que causen cualquier tipo de problema o disfuncionalidad. Por lo que estos son tomados en cuenta en la escritura del programa.
* Los comentarios dentro del código fuente del programa ayudan a que personas que no están familiarizadas con el código, pueden entender cómo funciona, y más adelante poder hacer una contribución al mismo.
* Las condiciones deben de estar correctamente redactadas y los valores incluidos dentro de estos deben hacer correcto uso de los operadores junto con su tipo específico, para que esta funcionen correctamente.
* Pudimos observar que es importante predefinir el valor de una variable, especialmente a la hora de usar contadores, o en este caso el de los elevadores, para que al operar con los datos no exista ningún tipo de error.

**BIBLIOGRAFIA:**

* Juego en el que se base el programa <http://www.engineering.com/GamesPuzzles/LogicElevators/tabid/4701/Default.aspx>